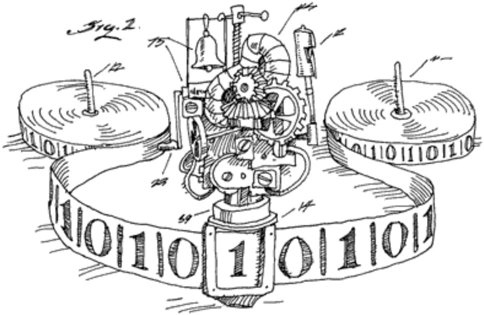
Superturingmaschinen

Ingo Blechschmidt

Curry Club Augsburg

4. Oktober 2016 und 3. November 2016

Ein Hoch auf Turingmaschinen



1. Schlichtheit
2. Mechanischer Bezug
3. Robustheit des Konzepts
4. Äquivalenz zu anderen Modellen
5. Querverbindungen

Lustiges zu Turingmaschinen

1. Schon kleine Turingmaschinen sind diffizil.
2. Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.

Lustiges zu Turingmaschinen

1. Schon kleine Turingmaschinen sind diffizil.
2. Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
3. Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \to \mathbb{N}$ überein.

Lustiges zu Turingmaschinen

1. Schon kleine Turingmaschinen sind diffizil.
2. Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
3. Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \to \mathbb{N}$ überein.
4. Eine Menge ist genau dann rekursiv aufzählbar, wenn sie durch eine Σ_1-Aussage definierbar ist:

$$\{n \in \mathbb{N} \mid \text{es gibt } m \in \mathbb{N} \text{ mit } \heartsuit\},$$

Lustiges zu Turingmaschinen

1. Schon kleine Turingmaschinen sind diffizil.
2. Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
3. Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \to \mathbb{N}$ überein.
4. Eine Menge ist genau dann rekursiv aufzählbar, wenn sie durch eine Σ_1-Aussage definierbar ist:

$$\{n \in \mathbb{N} \,|\, \text{es gibt } m \in \mathbb{N} \text{ mit } \heartsuit\},$$

und wenn sie diophantisch ist:

$$\{n \in \mathbb{N} \,|\, \text{die Gl. } f(n, x_1, \ldots, x_m) = 0 \text{ besitzt eine Lösung}\},$$

wobei f ein Polynom mit ganzzahligen Koeffizienten ist.

Eine Σ_1-Aussage ist eine Aussage der Form

$$\text{„Es gibt } m \in \mathbb{N} \text{ mit } \heartsuit.\text{“},$$

wobei in der Teilaussage $\heartsuit$ nur noch *beschränkte Quantifikation* vorkommen darf – also Formeln wie

$$\text{„Für alle Zahlen kleiner als } \cdots \text{ gilt } \ldots\text{“}$$

oder

$$\text{„Es gibt eine Zahl kleiner als } \cdots \text{ mit } \ldots\text{“}$$

und nicht Formeln wie

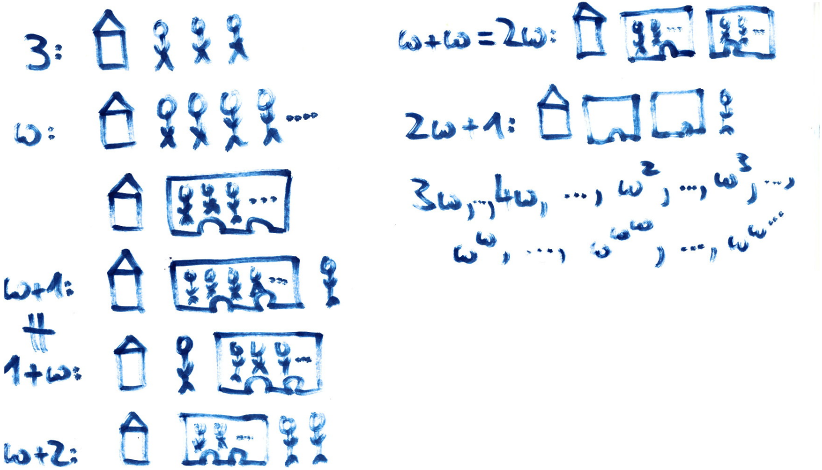
$$\text{„Für alle Zahlen gilt } \ldots\text{“}$$

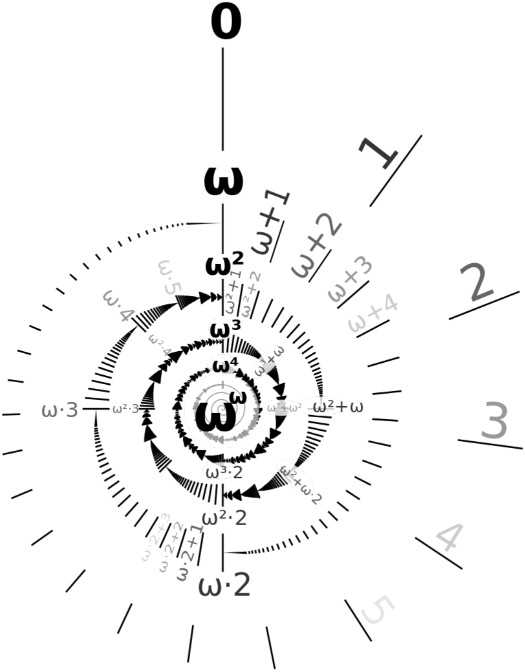
und

$$\text{„Es gibt eine Zahl mit } \ldots\text{“}.$$

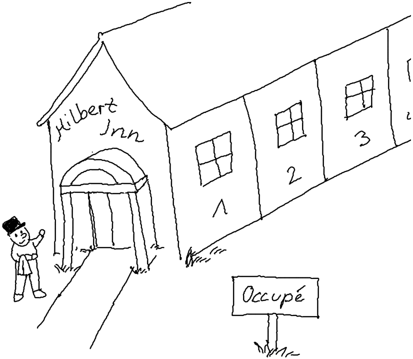
Die Teilaussage $\heartsuit$ muss also in endlicher Zeit überprüfbar sein.

Ordinalzahlen messen Anordnung



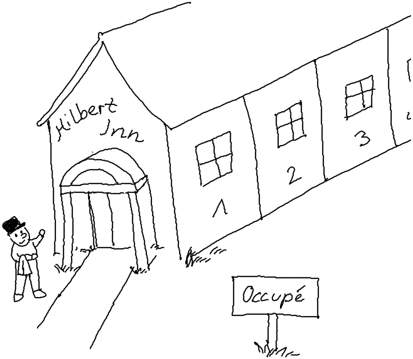

0
ω
1
$\omega+1$
$\omega+2$
2
$\omega+3$
$\omega+4$
ω^2
$\omega\cdot5$
$\omega\cdot4$
ω^2+1
ω^2+2
ω^3
$\omega^2\cdot4$
ω^4
$\omega^3+\omega$
$\omega\cdot3$
$\omega^2\cdot3$
ω^ω
$\omega^3+\omega^2$
$\omega^2+\omega$
3
$\omega^3\cdot2$
$\omega^2+\omega\cdot2$
$\omega^2\cdot2$
4
$\omega\cdot2+3$
$\omega\cdot2+2$
$\omega\cdot2+1$
$\omega\cdot2$
5

Kardinalzahlen messen Anzahl



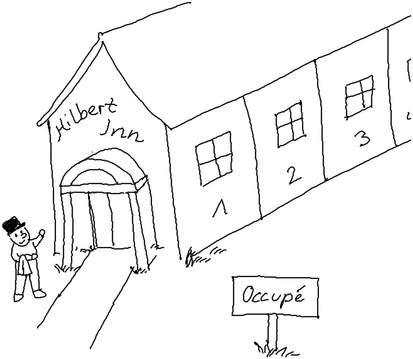
- Es gibt $\aleph_0$ viele natürliche Zahlen.

Kardinalzahlen messen Anzahl



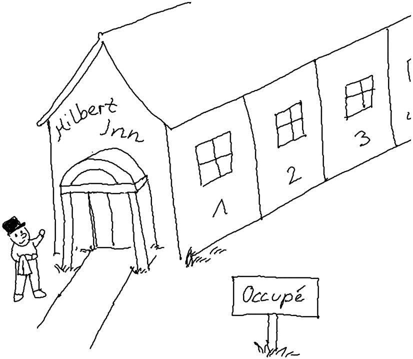
- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$,

Kardinalzahlen messen Anzahl



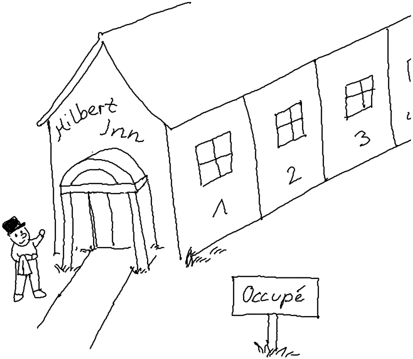
- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0, \qquad \aleph_0 + \aleph_0 = \aleph_0,$

Kardinalzahlen messen Anzahl



- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0, \quad \aleph_0 + \aleph_0 = \aleph_0, \quad \aleph_0 \cdot \aleph_0 = \aleph_0.$

Kardinalzahlen messen Anzahl



- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0, \quad \aleph_0 + \aleph_0 = \aleph_0, \quad \aleph_0 \cdot \aleph_0 = \aleph_0.$
- Es gibt mehr als $\aleph_0$ viele reelle Zahlen.

Was sind Superturingmaschinen?

Bei Superturingmaschinen ist die Zeitachse spannender:

- normal: $0,\ 1,\ 2,\ \ldots$
- super: $\quad 0,\ 1,\ 2,\ \ldots,\ \omega,\ \omega+1,\ \ldots,\ 2\omega,\ 2\omega+1,\ \ldots\ldots\ldots\ldots$

Wird eine Limesordinalzahl erreicht, so wird …

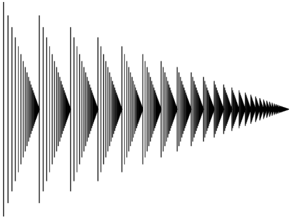
Was sind Superturingmaschinen?

Bei Superturingmaschinen ist die Zeitachse spannender:

- normal: $0,\ 1,\ 2,\ \ldots$
- super: $0,\ 1,\ 2,\ \ldots,\ \omega,\ \omega+1,\ \ldots,\ 2\omega,\ 2\omega+1,\ \ldots\ldots\ldots\ldots$

Wird eine Limesordinalzahl erreicht, so wird …

- die Maschine in einen designierten Zustand versetzt,
- der Schreib-/Lesekopf auf den Anfang bewegt und
- der „lim sup“ aller vorherigen Bandinhalte genommen.



Was können Superturingmaschinen?

- Alles, was gewöhnliche Turingmaschinen können.
- Zahlentheoretische Behauptungen überprüfen:
  - $\forall$ – „Für alle Zahlen gilt …“
  - $\exists$ – „Es gibt eine Zahl mit …“
  - $\forall\exists$ – „Für alle Zahlen n gibt es jeweils eine Zahl m mit …“
  - $\exists\forall$ – „Es gibt eine Zahl n, sodass für alle Zahlen m gilt: …“
  - $\forall\exists\forall$, $\exists\forall\exists$, …
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen simulieren.
- Π^1_1- und Σ^1_1-Aussagen entscheiden.

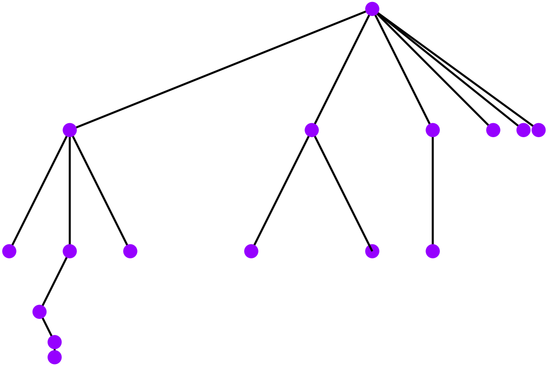
Was können Superturingmaschinen?

- Alles, was gewöhnliche Turingmaschinen können.
- Zahlentheoretische Behauptungen überprüfen:
  - $\forall$ – „Für alle Zahlen gilt …“
  - $\exists$ – „Es gibt eine Zahl mit …“
  - $\forall\exists$ – „Für alle Zahlen n gibt es jeweils eine Zahl m mit …“
  - $\exists\forall$ – „Es gibt eine Zahl n, sodass für alle Zahlen m gilt: …“
  - $\forall\exists\forall$, $\exists\forall\exists$, …
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen simulieren.
- Π^1_1- und Σ^1_1-Aussagen entscheiden.

Aber: Superturingmaschinen können nicht alle Funktionen berechnen und nicht jede 0/1-Folge aufs Band schreiben.

Fundierung von Bäumen

Ein Baum ist genau dann **fundiert**, wenn er keinen unendlichen Pfad enthält.



Superturingmaschinen können die Fundiertheit von Bäumen entscheiden.

Ein kleines Wunder

Superturingmaschinen können Π^1_1- und Σ^1_1-Aussagen entscheiden:

„Für jede Funktion $\mathbb{N} \to \mathbb{N}$ gilt …“

„Es gibt eine Funktion $\mathbb{N} \to \mathbb{N}$ mit …“

Und das, obwohl es überabzählbar viele Funktionen $\mathbb{N} \to \mathbb{N}$ gibt, aber Superturingmaschinen nur ein abzählbares Band verwenden und (nächste Folie) immer schon nach abzählbar vielen Schritten halten oder in Endlosschleifen geraten.

Wann halten Superturingmaschinen?

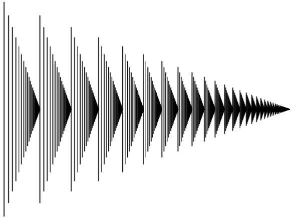
Schon nach **abzählbar vielen** ($\leq \aleph_0$ vielen) Schritten hält jede Superturingmaschine entweder an oder wiederholt sich.

Wann halten Superturingmaschinen?

Schon nach **abzählbar vielen** ($\leq \aleph_0$ vielen) Schritten hält jede Superturingmaschine entweder an oder wiederholt sich.

Sprechweise. Eine Ordinalzahl ist genau dann **abzählbar**, wenn sie nur abzählbar viele Vorgänger hat.

Genau die abzählbaren Ordinalzahlen lassen sich in $\mathbb{R}$ einbetten.



Notation. Es ist ω_1 die erste Ordinalzahl, vor der *überabzählbar* unendlich viele Ordinalzahlen kommen.

Behauptung. Hat eine Superturingmaschine nach *abzählbar vielen* Schritten noch nicht angehalten, so hält sie nie.

Beweis. Angenommen, eine Superturingmaschine hat vor Schritt ω_1 noch nicht gehalten. Dann gibt es eine Ordinalzahl $\alpha_0 < \omega_1$, zu der sich alle Zellen, die sich bis vor ω_1 stabilisieren werden, schon stabilisiert haben. Ferner gibt es Ordinalzahlen

$$\alpha_0 < \alpha_1 < \alpha_2 < \cdots < \omega_1,$$

sodass sich zwischen α_n und α_{n+1} all die Zellen, die sich bis ω_1 noch ändern werden, jeweils mindestens einmal ändern. Sei $\delta = \lim_{n\to\infty} \alpha_n$. Das ist eine Ordinalzahl $< \omega_1$, also eine abzählbare Ordinalzahl. Dann ist die Aufnahme der Superturingmaschine bei δ gleich der bei ω_1. Die Superturingmaschine wiederholt sich also.

Ausbrechen aus Wiederholungen

Was macht folgende Superturingmaschine?

Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.

- Wenn ja, dann halte.
- Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Ausbrechen aus Wiederholungen

Was macht folgende Superturingmaschine?

> Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.
>
> - Wenn ja, dann halte.
> - Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Sie scheint sich zu wiederholen, hält aber nach Schritt ω^2.

Ausbrechen aus Wiederholungen

Was macht folgende Superturingmaschine?

Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.

- Wenn ja, dann halte.
- Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Sie scheint sich zu wiederholen, hält aber nach Schritt ω^2.

Eine Superturingmaschine wiederholt sich genau dann, wenn

- die Aufnahmen zu zwei Limesordinalzeiten gleich sind und
- zwischen diesen Zeiten keine Zellen, die Null waren, zu Eins werden.

Stempelbare Ordinalzahlen

Eine Ordinalzahl α ist genau dann **stempelbar** (clockable), falls es eine Superturingmaschine gibt, die genau nach Schritt α hält.

- Jede endliche Ordinalzahl ist stempelbar.
- Stempelbar sind außerdem: ω, 2ω, ω^2
- Sind α und β stempelbar, so auch $\alpha + \beta$ und $\alpha \cdot \beta$.
- Nur abzählbar viele Ordinalzahlen sind stempelbar.
- Jede rekursive Ordinalzahl ist stempelbar.

Stempelbare Ordinalzahlen

Eine Ordinalzahl α ist genau dann **stempelbar** (clockable), falls es eine Superturingmaschine gibt, die genau nach Schritt α hält.

Beschleunigungssatz

Ist $\alpha + n$ stempelbar, so auch α.

Große-Lücken-Satz

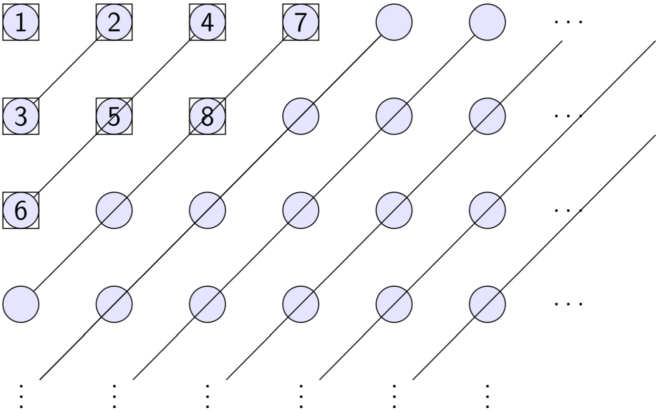
Für jede stempelbare Ordinalzahl α gibt es eine Lücke der Länge $\geq \alpha$ in den stempelbaren Ordinalzahlen.

Viele-Lücken-Satz

Ist α eine schreibbare Ordinalzahl, so gibt es mindestens α viele Lücken der Länge $\geq \alpha$ in den stempelbaren Ordinalzahlen.

Kurioserweise gibt es auch den *Lückenlose-Blöcke-Satz* (Gapless Blocks Theorem): Es gibt in den Ordinalzahlen „lange Abschnitte“ von lauter stempelbaren Ordinalzahlen.

Erinnerung: Diagonalisierung



Lückenexistenzsatz

Die erste Lücke nach jeder stempelbaren Ordinalzahl hat Länge ω.

Beweis. Sei α eine stempelbare Ordinalzahl. Sei β die kleinste nicht-stempelbare Ordinalzahl nach α. Dann gibt es keine stempelbaren Ordinalzahlen zwischen β und $\beta + \omega$. Und $\beta + \omega$ selbst ist stempelbar durch folgendes Programm:

> Simuliere alle Superturingmaschinen auf verzahnte Art und Weise. Behalte dabei insbesondere das Programm im Auge, das nach Schritt α halten wird. Sobald dieses gehalten hat, simuliere so lange weiter, bis der Zeitpunkt β erreicht ist, zu dem keine Superturingmaschine hält, und halte dann.

Zur Erkennung waren aber noch ω Schritte nötig.

Lost-Melody-Theorem

Es gibt Bandinhalte, die

- Superturingmaschinen nicht schreiben, aber
- erkennen können.

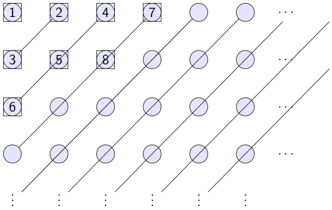
Lost-Melody-Theorem

Es gibt Bandinhalte, die

- Superturingmaschinen nicht schreiben, aber
- erkennen können.

Beweis. Sei c eine Kodierung aller Ablauffolgen aller Superturingmaschinen als unendliche 0/1-Folge.

- Dann ist c nicht schreibbar.
- Aber c ist erkennbar.



Mathematische Alternativuniversen

Zu jedem Rechenmodell $\mathcal{M}$ gibt es einen **Topos** $\mathrm{Eff}(\mathcal{M})$, in den wir mit **Realisierbarkeitstheorie** hineinschauen können.

Mathematische Alternativuniversen

Zu jedem Rechenmodell $\mathcal{M}$ gibt es einen **Topos** $\mathrm{Eff}(\mathcal{M})$, in den wir mit **Realisierbarkeitstheorie** hineinschauen können.

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Zahl n gibt es eine Primzahl $p > n$."
bedeutet:
Es gibt eine Turingmaschine, die eine Zahl n vom Band einliest und eine Primzahl $p > n$ als Ausgabe aufs Band schreibt.

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Zahl besitzt eine Primfaktorzerlegung."
bedeutet:
Es gibt eine Turingmaschine, die eine Zahl n vom Band einliest und eine Liste von Primzahlen, deren Produkt n ist, aufs Band schreibt.

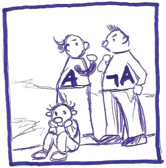
Wenn man ganz normal Mathematik betreibt, so arbeitet man im „Topos der Mengen“. Es gibt aber auch andere mathematische Universen neben diesem. In diesen anderen Topoi gelten teilweise leicht andere logische Gesetze als die uns vertrauten; somit sieht die mathematische Landschaft in manchen Topoi etwas anders aus als die bekannte Landschaft im Topos der Mengen.

Jede geometrische Form (jeder topologischer Raum, jeder „Situs“) gibt Anlass zu jeweils einem Topos (den „Topos der Garben über der Form“); die entstehenden Topoi heißen „Grothendieck-Topoi“, da es Grothendieck war, der dieses Konzept in den 1960er Jahren einführte. Topoi anderer Art sind die effektiven Topoi, die es für jeden Rechenmodell gibt: gewöhnliche Turingmaschinen (TM), Superturingmaschinen (STM), untypisiertes λ-Kalkül (λC), Maschinen in der realen Welt (RW), …

Neben der logischen Sicht gibt es auch eine geometrische Sicht auf Topoi: Topoi kann man sich als verallgemeinerte Arten von Räumen vorstellen. Etwa gibt es das Konzept des *Punkts* eines Topos, das von offenen, abgeschlossenen und dichten *Untertopoi* und das von stetigen Abbildungen zwischen Topoi.

Was gilt in Alternativuniversen?

Metatheorem: Jede Aussage, die sich **intuitionistisch** beweisen lässt, gilt in allen Topoi.



Schon gewusst?

Intuitionistische Logik ist wie klassische Logik, nur ohne:

- Axiom vom ausgeschlossenen Dritten (LEM): $\varphi \vee \neg\varphi$
- Axiom der Doppelnegationselimination (DNE): $\neg\neg\varphi \Rightarrow \varphi$

So sind Widerspruchsbeweise nicht pauschal möglich.

Wertschätzung intuitionistischer Logik

Der Verzicht aufs Axiom vom ausgeschlossenen Dritten erscheint zunächst schade, ist es doch ein gelegentlich hilfreiches Axiom. Allerdings:

- Man benötigt das Axiom seltener, als man vielleicht denkt.
- Der Verzicht ist gut für die mentale Hygiene.
- Mit intuitionistischer Logik kann man feinere Unterschiede abbilden.
- Aus intuitionistischen Beweisen kann man stets Programme extrahieren, die die bewiesenen Behauptungen bezeugen.
- **Der Verzicht auf LEM ermöglicht die Hinzunahme kurioser unkonventioneller Axiome.**

In klassischer Logik ist man gewohnt, doppelte Verneinungen sofort wegzustreichen. Da das in intuitionistischer Logik nicht möglich ist, muss man etwas mehr sprachliche Sorgfalt walten lassen. Insbesondere muss man zwischen echten Widerspruchsbeweisen und Beweisen von negierten Aussagen unterscheiden, in denen ebenfalls das Wort „Widerspruch" vorkommt:

- *Beweis einer negierten Aussage (intuitionistisch zulässig):* Wir möchten $\neg\psi$ zeigen. Angenommen, die Aussage ψ würde stimmen. Dann …, das wäre ein Widerspruch. Somit $\neg\psi$.

  In intuitionistischer Logik (wie auch in klassischer Logik) ist $\neg\psi$ nämlich als die Implikation $(\psi \Rightarrow \bot)$ definiert.

- *Beweis durch Widerspruch (intuitionistisch nicht pauschal möglich):* Angenommen, die zu zeigende Aussage φ wäre falsch. Dann …, das wäre ein Widerspruch. Somit ist φ wahr.

  Das intuitionistisch zulässige Teil in diesem Argument zeigt nur, dass φ *nicht nicht* gilt: $\neg\neg\varphi$.

Vor ein paar Jahren tauchte ein Video von Kate Moss auf, das sie beim Konsumieren von Drogen zeigte. Aus dem Video war klar, dass es sich dabei entweder um Drogen von einem gewissen Typ A oder von einem gewissen Typ B handelte; aber es gab keinen direkten Beleg für einen der beiden Typen. Kate Moss wurde nicht verfolgt; in diesem Sinn verwendete das Justizsystem also intuitionistische Logik. Siehe ein Blog-Post von Dan Piponi über das Thema.

Intuitionistische Logik kann feinere Unterschiede abbilden als klassische Logik. Wenn wir zum Beispiel wissen, dass sich unser Haustürschlüssel irgendwo in der Wohnung befinden muss (da wir ihn letzte Nacht verwendet haben, um die Tür aufzusperren), wir ihn momentan aber nicht finden, so können wir intuitionistisch nicht die Aussage vertreten, dass es eine Stelle gäbe, an der der Schlüssel liege. Denn dazu müssten wir in der Lage sein, einen expliziten Zeugen dieser Existenzbehauptung (also den Aufenthaltsort des Schlüssels) anzugeben. Wir können intuitionistisch nur die durch doppelte Negation abgeschwächte Aussage vertreten. (Die Beispiele hinken etwas, da sie sich auf den Kenntnisstand von gewissen Personen beziehen.)

Nebenbei bemerkt: Intuitionistische Mathematikerinnen behaupten *nicht*, dass das Axiom vom ausgeschlossenen Dritten falsch sei (d. h. dass seine Negation gelten würde). Sie setzen es nur nicht in seiner vollen Allgemeinheit voraus.

Tatsächlich lassen sich manche Instanzen des Axioms auch intuitionistisch zeigen: Zum Beispiel folgt mit Induktion, dass jede natürliche Zahl gleich Null oder ungleich Null ist.

Die analoge Behauptung für reelle Zahlen lässt sich intuitionistisch nicht zeigen. Das hat eine Parallele in der Programmierung: Bekanntlich ist es unschicklich, Fließkommazahlen auf Gleichheit zu testen, während das bei Ganzzahlen kein Problem ist. Das wird auf einer späteren Folie noch näher ausgeführt.

Eine letzte Bemerkung. Philosophinnen untersuchen nicht nur was *wahr* ist, sondern auch was stimmen *sollte*, was *möglich* ist, was *notwendig* ist, was jemand *weiß* oder was jemand *glaubt*. Das formalisiert man mit *modalen Operatoren.*

Als Mathematikerin kann man daher manchmal neidisch werden. In intuitionistischer Logik aber gibt es durchaus auch eine Vielzahl von modalen Operatoren! Die Doppelnegation ist das wichtigste Beispiel.

LEM für Gleichheit von Funktionen

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \to \mathbb{N}$ gilt: Entweder ist f die Nullfunktion oder nicht.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann entscheidet, ob M stets Null als Ausgabe produziert oder nicht.

Das stimmt nicht.

LEM für Gleichheit von Funktionen

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \to \mathbb{N}$ gilt: Entweder ist f die Nullfunktion oder nicht."

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann entscheidet, ob M stets Null als Ausgabe produziert oder nicht.

Das stimmt nicht.

Bemerkung: Aus externer Sicht ist das Objekt $\mathbb{N}^{\mathbb{N}}$ des effektiven Topos die Menge der **berechenbaren** Funktionen $\mathbb{N} \to \mathbb{N}$.

LEM für Gleichheit von Funktionen

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \to \mathbb{N}$ gilt: Entweder ist f die Nullfunktion oder nicht.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann entscheidet, ob M stets Null als Ausgabe produziert oder nicht.

Das stimmt nicht.

Bemerkung: Aus externer Sicht ist das Objekt $\mathbb{N}^{\mathbb{N}}$ des effektiven Topos die Menge der **berechenbaren** Funktionen $\mathbb{N} \to \mathbb{N}$.

In $\mathrm{Eff}(\mathrm{STM})$ stimmt die Aussage.

LEM fürs Halten von Turingmaschinen

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Turingmaschine M hält oder hält nicht."

bedeutet:

Es gibt eine Turingmaschine, die die Kodierung einer Turingmaschine M als Eingabe vom Band liest und dann entscheidet, ob M hält oder nicht.

Das stimmt nicht.

LEM fürs Halten von Turingmaschinen

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Turingmaschine M hält oder hält nicht.“
bedeutet:

Es gibt eine Turingmaschine, die die Kodierung einer Turingmaschine M als Eingabe vom Band liest und dann entscheidet, ob M hält oder nicht.

Das stimmt nicht.

In $\mathrm{Eff}(\mathrm{STM})$ stimmt die Aussage.

LEM für Gleichheit reeller Zahlen

Die Aussage

„Jede reelle Zahl ist Null oder ungleich Null."

ist in intuitionistischer Logik äquivalent zu

„Jede Turingmaschine hält oder hält nicht.",

gilt also nicht in $\mathrm{Eff}(\mathrm{TM})$, aber in $\mathrm{Eff}(\mathrm{STM})$.

Denn zu einer Turingmaschine M kann man die reelle Zahl $0{,}000\ldots$ betrachten, deren n-te Nachkommastelle genau dann eine 1 ist, wenn M nach Schritt n hält, und sonst 0 ist.

Umgekehrt kann man zu einer reellen Zahl x diejenige Turingmaschine betrachten, die die Nachkommaziffern von x nach einer von Null verschiedenen Ziffer absucht und genau dann hält, wenn sie eine solche Ziffer gefunden hat.

Stimmt die Aussage in $\mathrm{Eff}(RW)$, dem effektiven Topos zu Maschinen der realen Welt? Das heißt: Ist es möglich, in der realen Welt ein Halteorakel zu bauen? Also eine Maschine, welche die Beschreibung einer Turingmaschine einliest und dann korrekt ausgibt, ob die Turingmaschine hält oder nicht?

Da es hier nicht um das Halteproblem für Maschinen der realen Welt geht (welches für Maschinen der realen Welt wegen des üblichen Arguments nicht entscheidbar ist), sondern nur um das Halteproblem für Turingmaschinen, ist die Antwort „ja“ nicht prinzipiell ausgeschlossen. Vielleicht sind Tricks mit schwarzen Löchern und relativistischer Zeitdilatation möglich.

Markovs Prinzip

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \to \mathbb{N}$, welche nicht die Nullfunktion ist, gibt es eine Stelle $n \in \mathbb{N}$ mit $f(n) \neq 0$.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ und zwar nicht die Nullfunktion berechnet, als Eingabe vom Band liest und dann eine Zahl n aufs Band schreibt, sodass M bei Eingabe von n nicht Null aufs Band schreibt.

Das stimmt! (Unbeschränkte Suche.)

Das Auswahlaxiom

Eine Menge X ist genau dann **projektiv**, wenn für jede Menge Y und jede Aussage $\varphi(x, y)$ mit Parametern $x \in X$, $y \in Y$ gilt:

> Wenn es zu jedem $x \in X$ ein $y \in Y$ mit $\varphi(x, y)$ gibt, so gibt es eine Abbildung $f : X \to Y$ mit $\varphi(x, f(x))$ für alle $x \in X$.

Das **Auswahlaxiom** klassischer Mathematik besagt, dass dort jede Menge projektiv ist.

- In $\mathrm{Eff}(\mathrm{TM})$ ist $\mathbb{N}$ projektiv, $\mathbb{N}^{\mathbb{N}}$ aber nicht.
- In $\mathrm{Eff}(\mathrm{STM})$ sind $\mathbb{N}$ und $\mathbb{N}^{\mathbb{N}}$ projektiv.
- In $\mathrm{Eff}(\mathrm{RW})$ ist $\mathbb{N}^{\mathbb{N}}$ projektiv, falls Blackboxes möglich sind.

Auch intuitionistisch ist jede endliche Menge projektiv. Der Beweis, dass $X = \{1, \ldots, n\}$ projektiv ist, geht so: Nach Voraussetzung existiert für jede Zahl $i = 1, \ldots, n$ je mindestens ein Element $y_i \in Y$ mit $\varphi(i, y_i)$. Also ist $f : X \to Y$ mit $i \mapsto y_i$ eine Abbildung der gewünschten Art.

Der Beweis lässt sich allerdings nicht auf den Fall übertragen, dass die Menge X unendlich groß ist. Der Grund dafür ist etwas subtil. Es hängt damit zusammen, dass Beweise nicht unendlich lang sein dürfen und dass in einem Geltungsbereich (Scope) nur endlich viele Variablen vorkommen dürfen.

Dass die Aussage des Auswahlaxioms nicht trivial ist, zeigt sich, wenn man die Aussage „$\mathbb{N}^{\mathbb{N}}$ ist projektiv“ in $\mathrm{Eff}(\mathrm{TM})$ interpretiert. Aus

> Es gibt eine Turingmaschine M, die die Kodierung einer Turingmaschine P, welche eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, vom Band einliest und die Kodierung eines Elements $M(P) \in Y$ zusammen mit einem Zeugen von $\varphi(f, M(P))$ ausgibt.

folgt nämlich nicht, dass:

> Es gibt eine berechenbare Abbildung g von der Menge der berechenbaren Funktionen $f : \mathbb{N} \to \mathbb{N}$ in die Menge Y sodass für alle solche Funktionen f in $\mathrm{Eff}(\mathrm{TM})$ gilt, dass $\varphi(f, g(f))$.

Man könnte denken, dass die gesuchte berechenbare Abbildung g einfach von M berechnet wird. Aber das stimmt im Allgemeinen nicht. Denn ist $f : \mathbb{N} \to \mathbb{N}$ eine durch eine Turingmaschine P berechenbare Funktion, so hängt das Element $M(P)$ im Allgemeinen von der konkreten Umsetzung von f durch P ab. Verschiedene Umsetzungen können verschiedene Elemente geben. Bei einer Abbildung, wie es g werden soll, darf das nicht sein.

Man kann zeigen (schöne Übungsaufgabe): In $\mathrm{Eff}(\mathcal{M})$ ist eine Assembly X genau dann projektiv, wenn es eine $\mathcal{M}$-Maschine A gibt, die kanonische Darstellungen für alle Elemente von X berechnet: Ist r eine Kodierung eines Elements x von X, so ist auch $A(r)$ eine Kodierung von x; und ist s eine weitere Kodierung von x, so ist $A(r) = A(s)$.

- Für $X = \mathbb{N}^{\mathbb{N}}$ aus $\mathrm{Eff}(\mathrm{TM})$ gibt es keine solche Maschine.

- Für $X = \mathbb{N}^{\mathbb{N}}$ aus $\mathrm{Eff}(\mathrm{STM})$ gibt es eine solche Maschine: Lese die Kodierung einer Superturingmaschine R, welche eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, vom Band ein. Simuliere dann auf verzahnte Art und Weise alle Superturingmaschinen. Sobald eine gefunden ist, deren Ausgabeverhalten dem von R gleicht, stoppe und gebe die Kodierung der gefundenen Superturingmaschine aus.

- Für $X = \mathbb{N}^{\mathbb{N}}$ aus $\mathrm{Eff}(\mathrm{RW})$ gibt es eine solche Maschine, falls in der realen Welt Blackboxes möglich sind: Die gesucht Maschine nimmt eine Maschine der realen Welt entgegen und verpackt sie in eine Blackbox, sodass die Kommunikation mit der Maschine nur noch über ihr Interface möglich und ihre innere Funktionsweise unbeobachtbar ist.

Church–Turing-These

Die **Church–Turing-These** besagt: Lässt sich eine Funktion $f : \mathbb{N} \to \mathbb{N}$ in der „realen Welt berechnen", so gibt es eine Turingmaschine, die f berechnet.

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Funktion $f : \mathbb{N} \to \mathbb{N}$ lässt sich durch eine Turingmaschine berechnen."

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann die Kodierung einer Turingmaschine, welche f berechnet, aufs Band schreibt.

Das ist trivial! „cat" ist die gesuchte Maschine.

Church–Turing-These

Die **Church–Turing-These** besagt: Lässt sich eine Funktion $f : \mathbb{N} \to \mathbb{N}$ in der „realen Welt berechnen", so gibt es eine Turingmaschine, die f berechnet.

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Funktion $f : \mathbb{N} \to \mathbb{N}$ lässt sich durch eine Turingmaschine berechnen."

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann die Kodierung einer Turingmaschine, welche f berechnet, aufs Band schreibt.

Das ist trivial! „cat" ist die gesuchte Maschine.

In $\mathrm{Eff}(\mathrm{STM})$ und $\mathrm{Eff}(\lambda C)$ stimmt die Aussage nicht.

Der effektive Topos zu Turingmaschinen ist also eine schöne Umgebung für die Informatik, da in ihm *jede Funktion* $\mathbb{N} \to \mathbb{N}$ *berechenbar ist* und unberechenbare Funktionen aussortiert wurden.

Vielleicht vermutet man hierbei einen Widerspruch. Sind nicht etwa folgende Funktionen unberechenbar? Was verhindert ihre Existenz im effektiven Topos?

$$H(n) = \begin{cases} 1, & \text{falls die } n\text{-te Turingmaschine hält,} \\ 0, & \text{sonst.} \end{cases}$$

$$\begin{aligned} BB(n) = {} & \text{höchste Anzahl Schritte, die eine haltende Turingmaschine} \\ & \text{mit } n \text{ Zuständen durchführt, bevor sie hält.} \end{aligned}$$

Um zu zeigen, dass H und BB beides totale Funktionen $\mathbb{N} \to \mathbb{N}$ sind – und nur auf solche bezieht sich die Church–Turing-These –, ist LEM nötig! Für H, damit die Fallunterscheidung getroffen werden kann. Für BB, weil implizit das Lemma verwendet wurde, dass eine subendliche Menge natürlicher Zahlen ein größtes Element enthält.

Die Aussage, dass *jede* Funktion $\mathbb{N} \to \mathbb{N}$ durch eine Turingmaschine berechenbar ist, ist auch als *formale Church–Turing-These* bekannt.

- In Eff(STM) gilt sie nicht, denn aus einer Superturingmaschine, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, kann man nur in den seltensten Fällen in eine gewöhnliche Turingmaschine mit demselben Ausgabeverhalten umwandeln.

- In Eff(λC) gilt die formale Church–Turing-These ebenfalls nicht, aber aus einem anderen Grund. Die These lautet in diesem Fall:

  > Es gibt einen λ-Term T, sodass für jeden λ-Term u, welcher eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, der λ-Term $T\ u$ die Kodierung einer Turingmaschine ist, welche f berechnet.

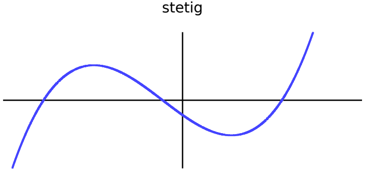
  Der ominöse Term T kann sein Argument u nur auf Eingaben auswerten, er hat aber keinen Zugriff auf die syntaktische Struktur von u. (Konventionsgemäß werden in Eff(TM) Turingmaschinen höherer Ordnung dadurch realisiert, dass sie als Argument Kodierungen von Turingmaschinen erhalten. Beim λ-Kalkül ist der Umweg über syntaktische Kodierungen nicht nötig und wird in Eff(λC) auch nicht verfolgt.)

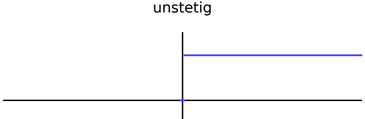
Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig."

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.





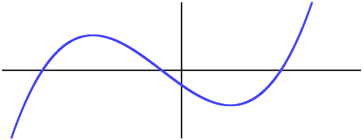
Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

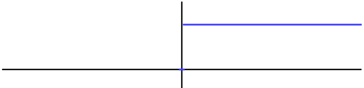
„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig."

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.

stetig



unstetig



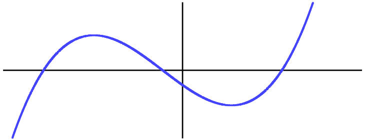
Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

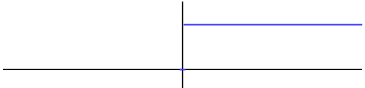
„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig."

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.

stetig



unstetig



Gibt es nicht offensichtlich unstetige Funktionen? Wie etwa die Signumfunktion?

$$\mathrm{sgn} : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \begin{cases} -1, & \text{falls } x < 0, \\ 0, & \text{falls } x = 0, \\ 1, & \text{falls } x > 0. \end{cases}$$

Ohne Verwendung von LEM ist die Situation nicht so trivial. Ohne es kann man nämlich nicht zeigen, dass diese Zuordnung eine auf ganz $\mathbb{R}$ definierte Funktion definiert: Dazu benötigte man das Lemma, dass jede reelle Zahl kleiner, gleich oder größer als Null ist. Dieses Lemma impliziert aber die schwächere Aussage, dass jede reelle Zahl gleich Null oder ungleich Null ist. Wir haben schon gesehen, dass diese Aussage in $\mathrm{Eff}(\mathrm{TM})$ nicht stimmt.

Ohne LEM definiert obige Zuordnung nur eine Funktion $M \to \mathbb{R}$, wobei $M = \{x \in \mathbb{R} \mid x < 0 \vee x = 0 \vee x > 0\}$. Über solche Funktionen geht es hier aber nicht.

Die in der Mathematik üblichen Definitionen des Konzepts reeller Zahlen sind in $\mathrm{Eff}(\mathrm{TM})$, $\mathrm{Eff}(\mathrm{STM})$ und $\mathrm{Eff}(\mathrm{RW})$ alle zueinander äquivalent (obwohl sie es unter bloßer Verwendung von intuitionistischer Logik nicht sind).

Aus externer Sicht ist eine reelle Zahl in jedem der drei Fälle durch eine Maschine gegeben, die in sich konsistente, beliebig genaue Approximationen produziert. Etwa kann man eine solche Maschine nach Approximationen auf drei, sieben und zehn Ziffern fragen und die Antworten

$$3.1417777777, \quad 3.1415926777 \quad \text{und} \quad 3.1415926535$$

erhalten.

Maschinen, welche andere, aber genauso gute Approximationen liefern, stellen dieselbe reelle Zahl dar.

Die Aussage „Alle Funktionen $\mathbb{R} \to \mathbb{R}$ sind stetig.“ des effektiven Topos bedeutet: Es gibt eine Maschine M, die

1. eine Maschine A, welche eine Funktion $f : \mathbb{R} \to \mathbb{R}$ berechnet,
2. eine Maschine X, welche eine reelle Zahl x repräsentiert, sowie
3. eine natürliche Zahl n

als Argumente nimmt und eine natürliche Zahl m mit folgender Eigenschaft ausgibt: Für jede reelle Zahl $\tilde{x}$, deren erste m Ziffern mit denen von x übereinstimmen, stimmen die ersten n Ziffern von $f(x)$ mit denen von $f(\tilde{x})$ überein.

Im Fall der realen Welt kann man wie folgt versuchen, eine derartige Maschine M zu konstruieren. Gegeben A, X und n, wendet sie A auf eine leichte Variante X' von X an: Die Maschine X' soll dasselbe Ausgabeverhalten zeigen wie X, allerdings bei jedem Aufruf über einen privaten Kommunikationskanal an M die Information übermitteln, wie viele Stellen abgefragt wurden. Da X' dasselbe Ausgabeverhalten wie X zeigt, muss A per Vertrag auf X' genauso reagieren wie auf X.

Wenn nun in endlicher Zeit nur endlich viele Rechenschritte durchführbar sind, so muss M nur warten, bis A die Zahl $f(x)$ auf n Stellen genau berechnet hat, und kann dann nachsehen, wie viele Stellen A für diese Berechnung benötigt hat. Für jede andere Zahl, die x in diesen Stellen gleicht, produziert somit A dieselben n Ziffern.

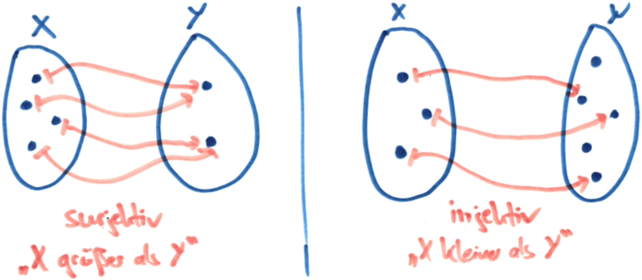
Kein Fan von reellen Zahlen? Dasselbe Phänomen zeigt sich auch bei Funktionen von anderen Typen. Zum Beispiel für Funktionen $f : \mathbb{B}^{\mathbb{N}} \to \mathbb{B}$. Dabei ist $\mathbb{B} = \{0, 1\}$ die Menge der Bools. Eine solche Funktion heißt genau dann *stetig*, wenn es für jedes Argument $x \in \mathbb{B}^{\mathbb{N}}$ (also jede Funktion $x : \mathbb{N} \to \mathbb{B}$) eine natürliche Zahl m gibt, sodass $f(x)$ nur von den ersten m Funktionswerten von x abhängt.

In $\mathrm{Eff}(\mathrm{TM})$ stimmt es, dass jede solche Funktion f stetig ist. Das ist der tiefere Grund dafür, wieso die „scheinbar unmöglichen Haskell-Programme“ funktionieren.

Seltsame Größenverhältnisse

Es gibt keine Surjektion $\mathbb{N} \to \mathbb{N}^\mathbb{N}$; die Menge $\mathbb{N}^\mathbb{N}$ der Funktionen $\mathbb{N} \to \mathbb{N}$ ist viel größer als $\mathbb{N}$.

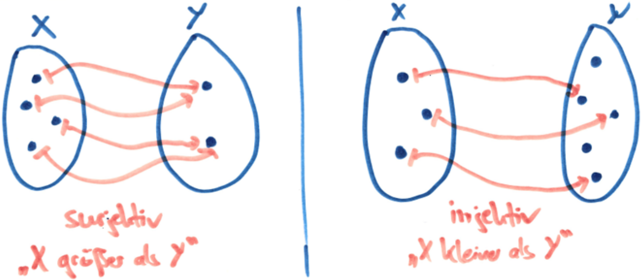
In klassischer Logik folgt: Es gibt auch keine Injektion $\mathbb{N}^\mathbb{N} \to \mathbb{N}$. Das drückt dieselbe Intuition über das Größenverhältnis aus.



Seltsame Größenverhältnisse

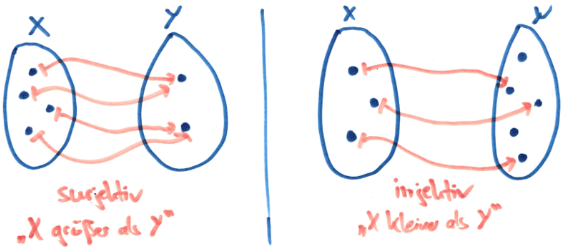
Es gibt keine Surjektion $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$; die Menge $\mathbb{N}^{\mathbb{N}}$ der Funktionen $\mathbb{N} \to \mathbb{N}$ ist viel größer als $\mathbb{N}$.

In klassischer Logik folgt: Es gibt auch keine Injektion $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. Das drückt dieselbe Intuition über das Größenverhältnis aus.



Aber in Eff(STM) gibt es eine solche Injektion!

Seltsame Größenverhältnisse



$\mathrm{Eff}(\mathrm{STM}) \models$ „Es gibt eine Injektion $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$."
bedeutet:

Es gibt eine Superturingmaschine, welche bei Eingabe einer Kodierung einer Superturingmaschine A, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, eine Zahl $n(A)$ berechnet und aufs Band schreibt. Dabei darf nur dann $n(A) = n(B)$ sein, wenn A und B dieselbe Funktion berechnen.

Seltsame Größenverhältnisse

Die Superturingmaschine

> Lese die Kodierung einer Superturingmaschine A vom Band ein. Simuliere nun alle Superturingmaschinen auf verzahnte Art und Weise. Sobald eine gefunden ist, deren Ausgabeverhalten dem von A gleicht, höre auf und gebe die Nummer n dieser Maschine aus.

schreibt bei Eingabe einer Kodierung einer Superturingmaschine A, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, eine Zahl $n(A)$ aufs Band. Dabei ist nur dann $n(A) = n(B)$, wenn A und B dieselbe Funktion berechnen.

Die von der beschriebenen Superturingmaschine ausgegebene Zahl n hängt vom Ausgabeverhalten von A, der gewählten Reihenfolge aller Superturingmaschinen und von den Details der Verzahnung ab – nicht aber von der Implementierung von A.

Die Suche terminiert, da es ja mindestens eine Superturingmaschine gibt, die bei Eingabe einer jeden natürlichen Zahl terminiert und dabei dasselbe Ausgabeverhalten wie A zeigt: A selbst.

Weil er so schön ist, hier der intuitionistisch zulässige Beweis, dass es keine Surjektion $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$ gibt:

Sei $s : \mathbb{N} \to \mathbb{N}^{\mathbb{N}}$ irgendeine Abbildung. Wir möchten nachweisen, dass s nicht surjektiv ist. Dazu betrachten wir die Abbildung $f : \mathbb{N} \to \mathbb{N}$ mit

$$f(n) := s(n)(n) + 1.$$

Dieses Element $f \in \mathbb{N}^{\mathbb{N}}$ kann von s nicht getroffen werden: Wenn es eine natürliche Zahl m mit $s(m) = f$ gäbe, so wäre

$$s(m)(m) = f(m) = s(m)(m) + 1.$$

Bei dieser Thematik ist es wichtig, nicht die externe und die interne Sicht durcheinanderzubringen. Sonst landet man bei *Skolems Paradox*: Wie kann es sein, dass es in $\mathrm{Eff}(\mathrm{TM})$ keine Surjektion $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$ gibt, wo doch $\mathbb{N}^{\mathbb{N}}$ nur berechenbare Funktionen enthält und es von diesen nur abzählbar viele gibt?

Die Auflösung ist: Es stimmt zwar, dass aus externer Sicht das Objekt $\mathbb{N}^{\mathbb{N}}$ von $\mathrm{Eff}(\mathrm{TM})$, das ist aus externer Sicht die Menge der berechenbaren Funktionen $\mathbb{N} \to \mathbb{N}$, abzählbar ist und dass es daher eine Surjektion (sogar Bijektion) $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$ gibt.

Diese Surjektion, und auch jede andere, ist aber entweder nicht berechenbar, daher nicht in $\mathrm{Eff}(\mathrm{TM})$ enthalten, oder zwar berechenbar, aber so beschaffen, dass es keinen berechenbaren Zeugen ihrer Surjektivität gibt. Daher sieht $\mathbb{N}^{\mathbb{N}}$ aus der Sicht von $\mathrm{Eff}(\mathrm{TM})$ wie eine überabzählbare Menge aus, in Übereinstimmung mit Cantors intuitionistisch zulässigen Beweis.